Git and GitHub Notes

Krystella Rattan

2/6/2022

Git commands for using GitHub

Summary

```
Command<-c("cd", "mkdir", "git init", "git add", "git add .", "git status", "git commit", "git commit -m

Description<-c("change directory", "make directory", "initialize repo", "add files to repo", "add all ch

summary_frame <- data.frame(Command, Description)

knitr::kable(summary_frame, "pipe", caption="**Running list of Git commands**", col.name=c("Command", "D
```

Table 1: **Running list of Git commands**

Command	Description
cd	change directory
mkdir	make directory
git init	initialize repo
git add	add files to repo
git add .	add all changes to the repo
git status	shows staging area
git commit	commit changes to the repo
git commit -m	commit changes and include a description
git checkout	move to another branch
git checkout -b	move to a new branch
git checkout master	move to the master branch
git remote add	add to a remote repo
git push -u	push files to a remote repo
git pull	pull files from a remote repo

Notes

The following notes were taken from https://www.youtube.com/watch?v=DVRQoVRzMIY

1. Designate the folder where you want the repository to be stored, e.g. Desktop
    a. Use the Git command **cd** to change this to the working directory

```
\users\username>cd Desktop
```

2. Create the repository folder using **mkdir**

```
\users\username\Desktop>mkdir test_repo
```

3. Change directory to the new repo folder

```
\users\username\Desktop>cd test_repo
```

4. Initialize the folder as a new Git repository using **git init**

```
\users\username\Desktop\test_repo>git init
```

This allows any changes made in this folder to be tracked by Git

5. Add files to the repo using **git add**

```
\users\username\Desktop\test_repo>git add readme.md
```

This adds files to the *"staging area"*, that is where we temporarily add or remove files that we want to be in our next commit

When several changes have been made, **git add .** can be used to add all changes

```
\users\username\Desktop\test_repo>git add .
```

6. Identify which files are currently in the staging area and what changes have been made, use **git status**

```
\users\username\Desktop\test_repo>git status
```

7. Commit files to the Git repository, use **git commit**

```
\users\username\Desktop\test_repo>git commit -m "added readme.md"
```

-m is used to add a short message describing what changes have been made

8. Create a new branch, use **git checkout -b**, or move across branches using **git checkout**

```
\users\username\Desktop\test_repo>git checkout -b branchname
```

The command git checkout -b *moves* you to a new branch, in this case 'branchname'
The command **git checkout master** moves you to the master branch

9. Add files to a *remote* repo, use **git remote add remote_repo_name url**

```
\users\username\Desktop\test_repo>git remote add origin <insert url copied from GitHub repo>
```

A remote is a url to another repository
The name of the remote repo (url) in this example was 'origin'

10. Push changes to a repo using **git push -u remote_repo_name branchname**

```
\users\username\Desktop\test_repo>git push -u origin master
```

-u origin - where to push to; master - which branch in the repo to push to

11. Pull changes from a remote repo using **git pull remote_repo_name branchname**

```
\users\username\Desktop\test_repo>git pull origin master
```

origin - remote repo name; master - which branch to pull from

12. Conflicts and merges

When changes are made to the same branch both locally and on the remote repository, pushing or pulling may result in a 'merge conflict'.

Git may not allow you to push to the remote repository until the changes have been manually resolved by assessing and editing the conflicting lines of code. The resolved file must then be saved locally, added to git (git add .), committed (git commit), and then pushed to the remote repository.

If you attempt to pull a branch into the repository that contains a conflict, you may be asked to compare before making a pull request. Using a 'draft pull request' can show the conflicts to be resolved before the merge can be made.